보안 버그 추적을 위한 파일 특징 분석

허진석*, 김영경**, 김미수**, 이은석**
*성균관대학교 정보통신대학
**성균관대학교 소프트웨어 대학
e-mail : {mrhjs225, agnes66, misoo12, leees} @skku.edu

Analyzing File Characteristic For Security Bug Localization

Jin-Seok Heo*, Young-Kyoung Kim**, Mi-Soo Kim**, Eun-Seok Lee**
*college of Information and Communication Engineering, Sung-Kyun-Kwan University
** Dept. of Software, Sung-Kyun-Kwan University

요 약

보안 버그는 소프트웨어의 치명적인 취약점을 노출해 제품의 질 저하 및 정보유출을 일으킨다. 위 상황을 최소화하기 위해 보안 버그 추적 기술이 필요하다. 본 논문에서는 보안 버그가 발생한 소스 파일의 특징을 분석하여 보안 버그 추적을 위한 정보를 제공한다. 우리는 보안이 중요하게 다루어져야 하는 안드로이드와 블록체인 오픈소스를 대상으로 보안 버그 리포트를 수집해 보안 버그가 나타난 소스 파일의 텍스트를 분석했다. 분석 결과, 안드로이드의 경우 통신 관련 패키지에 포함된 파일에서 보안 버그가 발생했다. 블록체인의 경우 계정, 키 저장 관련 파일들에서 보안 버그가 주로 발생했다. 보안 버그 추적 시 본 연구의 분석 결과를 반영한다면 빠르고 정확하게 보안 버그 파일을 찾을 수 있을 것으로 보인다.

1. 서론

최근 몇 년간 금융 관련 기관의 보안 사고가 잇달아 발생하면서 개인정보 유출 및 가상 화폐가 유출된 사건들이 큰 이슈이다[1]. 이로 인해 현대 사회에서 보안 버그의 중요성은 갈수록 높아지고 있다. 보안 버그는 버그 리포트들을 통해 개발자들에게 알려질 수 있다. 개발자들은 보안 버그 리포트를 읽고 이를 해결하기 위한 소스 파일을 찾아 버그를 수정한다. 이때 버그가 있는 파일을 찾는 데 소모되는 시간을 줄이기 위해 버그 추적 기술이 사용될 수 있다.

버그 추적 기술은 버그 리포트에 작성된 정보들을 바탕으로 해당 버그를 발생시킨 소스 파일들을 찾아준다. 이 기술은 리포트를 자연어 전처리 후 준 지도 학습 분류기[2]와 같은 기계 학습을 통해 추적한다. 또한 텍스트 군집화 기법[3]을 추가하여 추적 성능을 높이고 있다. 위 기술을 보안 버그 추적에 적용하기 위해서는 보안 버그가 발생한 파일들의 특징을 분석하여 그 결과를 반영하는 것이 필요하다.

본 연구는 보안 버그를 추적하기 위해 보안 버그가 나타난 소스 파일들의 특징을 분석한다. 이를 위해 보안이 중요한 안드로이드 및 블록체인 소프트웨어에 초점을 맞춰 텍스트 분석을 수행한다. 분석 결과를 이용해 보안 버그 추적 기술에 반영하기 위한 방향을 제시한다.

본 논문의 구성은 다음과 같다. 2 장에서 기존 버그 추적 기술에 대해서 살펴본다. 3 장에서 분석 대상과 방법에 관해 설명하고, 4 장에서 보안 버그 파일들의 분석 결과를 설명한다. 5 장에서는 분석 결과를 통해 기존 기술을 보안 버그에 적용할 방안을 논의하고, 6 장에서 본 연구의 결론과 향후 연구를 설명한다.

2. 관련 연구

버그 추적 기술 연구는 다양한 정보들과 기술들을 사용하여 진행되어 왔다. Naresh et al.는 버그 리포트에 쓰인 자연어를 분류학적 용어 대응 기법으로 처리해 버그 파일을 찾는 기술을 제안했다[3]. Chao et al.는 소프트웨어의 올바른 실행과 잘못된 실행 사이에 술어 패턴 모델을 구축하고, 통계적 분석으로 버그를 추적하는 기술을 제안했다[4]. 최근에는 정보검색 기술을 적용한 추적 기술들이 제안되었다. Klaus et al.는 버그 리포트와 소스 파일 간 문서 유사도, 문서의 구조적 정보, 코드 변경 이력, 스택 트레이를 활용하는 기술을 제안했다[5]. Mohammad et al.는 정보검색 기술 적용 시 낮은 품질의 리포트에서도 버그 파일을 잘 찾을 수 있도록 쿼리 재구성 기반 기술을 제안했다[6].

기존 연구들은 버그의 종류는 고려하지 않은 기술들을 제안해 왔다. 우리는 기존 기술들을 보안 버그에 적용하기 위한 보안 버그 파일의 특징을 분석한다.

3. 분석 방법

3.1 분석 대상

우리는 보안 버그 파일의 분석을 위해 보안이 중요한 안드로이드와 가상 화폐 기반 기술인 블록체인의 소프트웨어 프로젝트에 버그 리포트와 버그 파일을 수집하여 분석한다.

프로젝트 선정을 위해, 안드로이드의 경우 GitHub[1]에서 인기 있는 상위 40 개의 안드로이드 프로젝트 중 해결된 이슈 리포트가 많은 7 개의 프로젝트와 사용자가 많이 사용하는 2 개의 프로젝트를 선정했다. 블록체인 도메인의 경우 기존의 블록체인 도메인에서 버그를 분석한 연구[7]가 분석한 프로젝트를 대상으로 선정하였다.

선정된 프로젝트에서 버그리포트를 선별하기 위해 해결된 이슈 중 “Bug”로 라벨링 된 이슈를 수집했다. 본 연구에서 분석할 보안 버그 리포트의 선정을 위해 오픈 소스 소프트웨어의 버그를 분류한 연구[8]를 참조하였다. 마지막으로 보안 버그 파일 분석을 위해 수정된 파일을 갖는 보안 버그 리포트들을 선정하였고 결과는 <표 1>과 같다.

총 3,940 개의 버그 리포트 중 437 개가 보안 버그 리포트로 분류되었으며 정답 파일을 가지고 있는 보안 버그 리포트는 256 개였다. 본 연구는 수정된 파일이 존재하는 256 개의 보안 버그 리포트를 대상으로 분석을 수행했다.

3.2 분석 방법

3.2.1 안드로이드

안드로이드의 경우 프로젝트들의 범주와 용도가 다양하기 때문에 파일 내 수정된 단어들의 공통된 특징을 파악하기가 어렵다. 따라서 소스 파일이 포함된 패키지 내 단어들을 분석한다.

분석을 위해 자바의 Jgit[2] 라이브러리를 사용하여 소스 파일이 포함된 패키지들의 문자열을 수집한다. 패키지 이름을 구분하는 “/”을 기준으로 단어를 분리하고 프로젝트의 이름이 포함되는 단어들을 제거한다.

마지막으로 추출된 단어들을 대상으로 단어 빈도수를 계산한다. 단어가 자주 나온다는 것은 관련 패키지에 보안 문제가 있다는 것이므로, 빈도수가 높은 상위 10 개의 단어를 선정한다.

분야	프로젝트 수	BR 개수	SBR 개수 (비율)	수정된 파일이 있는 SBR 수 (비율)
안드로이드	9	2,487	306 (12.3%)	182 (7.3%)
블록체인	9	1,453	131 (9.0%)	74 (5.1%)
합계	18	3940	437 (11.1%)	**256 (6.5%)**

<표 1> 대상 프로젝트 버그 리포트 분류 결과

3.2.2 블록체인

안드로이드와 동일하게 보안 버그리포트를 해결하기 위해 수정된 소스 파일의 단어들을 분석한다. Jgit 라이브러리를 사용하여 소스 파일의 수정된 코드 부분을 수집한다. 띄어쓰기를 기준으로 단어 단위로 분리하며, 문법 키워드를 제거한 후 단어 빈도수를 계산한다. 빈도수가 높은 상위 10 개 단어를 선정한다.

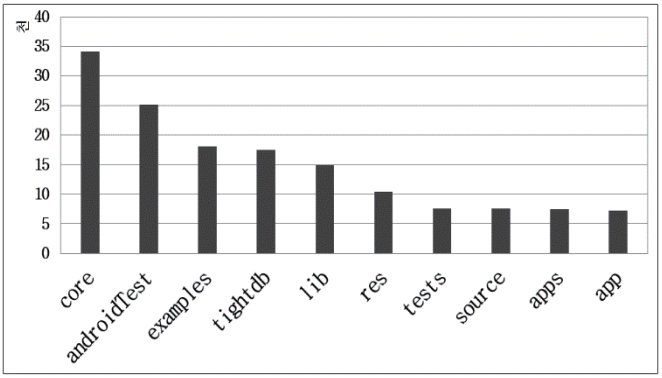
4. 분석 결과

4.1 안드로이드

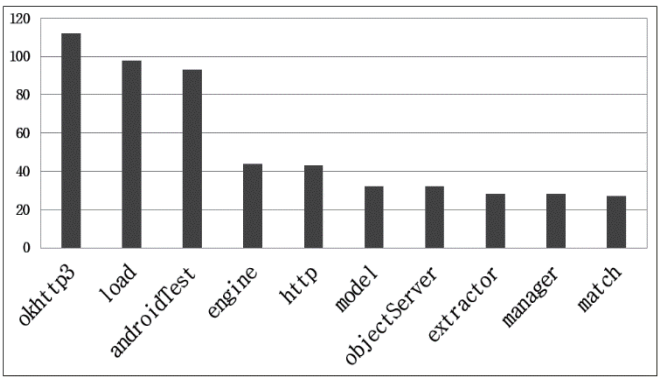
<그림 1>과 <그림 2>는 각각 안드로이드 분야에서 보안 버그가 아닌 패키지와 보안 버그가 나타난 패키지의 분석 결과이다.

보안 버그가 아닌 패키지를 단어 빈도수 순으로 나열 시 가장 많이 나오는 단어는 “core”로 34,098 회에 달했고, 상위 10 위인 단어는 “app”으로 7,262 회로 계산되었다. 보안 버그 관련 패키지들의 경우 “okhttp3”가 112 회로 가장 많이 등장했고, “match”이 상위 10 위로 27 회였다. 상위 10 개의 단어를 비교한 결과 두 종류의 버그가 발생하는 파일의 패키지는 “androidTest”를 제외하고 분명하게 다른 것을 확인하였다.

단어 빈도수를 분석한 결과 보안 버그와 관련된 패키지에는 통신 패키지가 주로 나타남을 알 수 있었다. 예시로 “okhttp3”, “load”, “http”와 같은 패키지가 존재했다.



<그림 1> 안드로이드 보안 버그 제외 패키지 단어 빈도수



<그림 2> 안드로이드 보안 버그 관련 패키지 단어 빈도수

[1] https://github.com/

[2] https://projects.eclipse.org/projects/technology.jgit

통신패키지의 경우 개인정보를 송수신하는 과정에서 예외 상황이나 혹은 앱의 크래시를 유발하는 경우가 많았다. 이 외에 SSL 인증거부 및 POST 방식 통신 실패들의 버그들이 있었다.

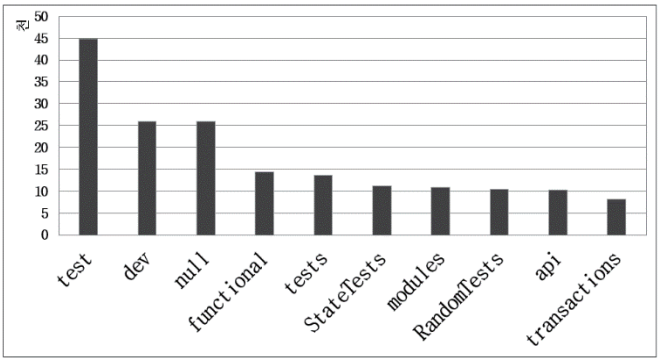
보안 버그가 아닌 리포트의 패키지 분석 결과, “androidTest”가 중복돼서 나타났다. 이 파일은 소프트웨어가 정상적으로 작동하는지 판단하는 테스트를 위한 파일이다. 이러한 과정에서 생기는 패키지 명이 “androidTest”, “Test”이다. 따라서 이는 버그와 관련이 없기 때문에 이 단어를 제외하고 분석하였다.

안드로이드의 보안 버그 파일을 분석한 결과, 보안 버그에서 나타나는 특징이 보안이 아닌 버그들에서 나타나지 않는다. 따라서 패키지 단어를 통해 보안 버그와 관련된 패키지를 구별할 가능성을 확인하였다.

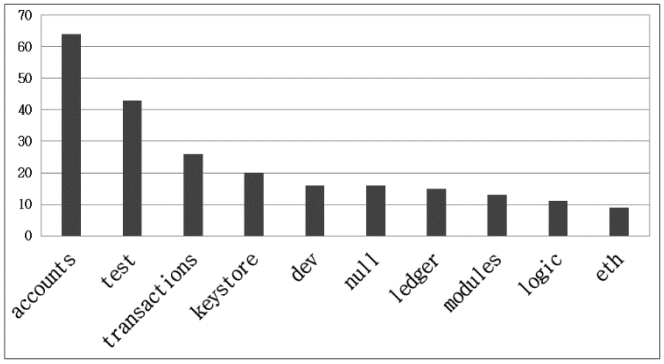
4.2 블록체인

<그림 3> 과 <그림 4>는 각각 블록체인 도메인에서 보안 버그가 아닌 버그들의 파일 분석 결과와 보안 버그 관련 파일 분석 결과이다.

블록체인의 경우 보안 버그가 아닌 파일의 경우 “test”가 44,833 회로 가장 큰 빈도수를 가졌고 “transactions”이 8,195 회로 상위 10 위의 빈도수를 가졌다. 보안 버그 관련 파일의 경우 “accounts”가 64 회로 가장 많이 등장했고, 상위 10 위에는 “eth”가 9 회의 빈도수를 보였다. 분석 결과를 통해 블록체인에서 “Account”, “Keystore” 관련 파일들이 보안 버그를 일으킴을 알 수 있었다.



<그림 3> 블록체인 보안 버그 버그 제외 단어 빈도수



<그림 4> 블록체인 보안 버그 관련 파일 단어 빈도수

“Account” 키워드를 갖는 소스 파일 들에서 보안 버그가 많이 발생했다. 개인신상 정보(ID, Password, 거래 명세)를 처리하는 종류의 파일들에 결함이 있었다. 이 경우 사용자가 주로 로그인, 로그아웃하는 데에 버그를 유발하며, 정확한 정보를 확인하지 않고 권한을 부여하는 데에 그 위험성이 있다.

“Keystore”와 관련하여, 주로 암호화된 키 혹은 API 키의 정보를 다루는 파일들이 많았다. 올바른 키 쌍을 비교하지 않아 프로그램이 깨지거나, 인증서를 통한 확인절차를 제대로 거치지 않아 버그를 일으키는 경우가 많았다. 이 외에 보안 버그와 관련이 있는 파일로는 SQL 문, SSL 통신 관련 파일들이 존재했다.

보안 버그가 아닌 파일의 분석 결과에서 “test”, “dev”, “null”, “modules”, “transactions”의 5 개 단어가 보안 버그 관련 파일의 분석 결과와 겹치는 것을 볼 수 있었다. 따라서 보안 버그에서 나타나는 특징들이 명확하다고 할 수 없다. 그러나 “accounts”, “keystore”를 포함한 5 개의 단어는 보안 버그가 아닌 파일의 분석 결과에는 등장하지 않았다.

따라서 부분적인 결과만을 이용한다면, 보안 버그를 유발하는 파일을 구별하는 것이 가능하다는 것을 확인할 수 있다.

5. 논의

본 연구의 분석결과와 프로젝트의 패키지 및 파일과의 대조를 통해 보안 버그가 있을 확률이 높은 파일을 추적할 수 있다. 그러나 오픈 소스의 프로젝트의 경우 프로젝트마다 같은 기능의 패키지이더라도 이름이 일치하지 않을 수 있다. 이 점을 보완하기 위해 보안 버그에 취약한 패키지 및 파일을 모아 보기 위한 ‘보안 인덱스(Security Index)’를 만든다.

안드로이드의 경우 통신 관련 패키지들을 표기하고, 블록체인의 경우 계정, 키 저장과 관련된 파일을 보안 인덱스에 표기하는 것을 제안한다. 기존 버그 추적 기술을 사용할 때 보안 인덱스를 추가한다면 인덱스에 표기된 파일들을 먼저 확인함으로써 보안 버그를 빠르게 추적할 수 있다.

블록체인의 도메인에 한해, 기존의 보안 버그 추적 기술에서 파일 내용을 분석할 때 본 연구의 분석 결과 “accounts”를 포함한 5 개의 단어가 많이 나오는 파일이 존재한다면 보안 버그가 있을 확률이 높은 파일이므로 보안 버그 추적을 위해 활용할 수 있다.

6. 결론

현재 보안 버그가 큰 이슈로 대두됨에 따라 신속한 수정이 필요해졌다. 따라서 기존 버그 추적 기술들을 보안 버그를 위해 활용할 방법을 연구해야 한다. 본 연구에서 보안 버그 추적 기술을 위해 보안 버그가 발생하는 소스 파일의 특징을 분석하였다.

안드로이드의 경우 통신과 리소스 관련 패키지가 높은 확률로 버그를 일으켰고, 블록체인의 경우 계정, 거래, 키 저장에 해당하는 세 가지 종류의 파일들이 보안 버그를 일으켰다. 위의 특징들은 보안 버그 추적 시 패키지 이름을 직접 비교함으로써 곧바로 사용되거나 보안 인덱스를 통하여 사용될 수 있다.

이번 연구에서 나온 결과를 버그 추적 기술에 반영하여 성능 향상의 정도를 측정하고, 보안 버그뿐만 아니라 영역을 넓혀 다른 종류의 버그 파일의 특징을 찾아 버그 추적 기술에 반영할 예정이다.

사사

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW 중심대학지원사업의 연구결과로 수행되었으며(2015-0-00914), 2018 년도 정부(교육부)의 재원으로 한국과학창의재단(2018 년도 학부생 연구프로그램)의 지원을 받아 수행된 연구임.

참고 문헌

[1] 심재홍, “금융 소비자를 위협하는 악성코드 위협사례 분석”, 한국인터넷진흥원, 5 월, 2013 년

[2] Jifeng Xuan, He Jiang, Zhilei Ren, Jun Yan and Zhongxuan Luo. “Automatic Bug Triage using Semi-Supervised Text Classification”, Proc. Of the 22nd International Conference on Software Engineering and Knowledge Engineering, pp.209-214, 2010.

[3] Naresh Kumar Nagwani and Shrish Verma. “CLUBAS : An Algorithm and JAVA based Tool for software bug classification using bug attributes similarities,” Journal of Software Engineering and Applications, Vol.5, No.6, pp.436-447, May, 2012.

[4] Chao Liu, Xifeng Yan, Long Fei, Jiawei Han and Samuel P. Midkiff. “SOBER : statistical model-based bug localization,” Proceedings of the 10th European Software Engineering Conference held jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering, pp.286-295, 2005.

[5] Klaus Changsun Youm, June Ahn and Eunseok Lee. “Improved bug localization based on code change histories and bug reports,” Information and Software Technology, Vol 82, pp.177-192, Feb, 2017.

[6] Mohammad Masudur Rahman and Chanchal K. Roy. “Improving IR-Based Bug localization with context-aware query reformulation,” The 26th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, 2018.

[7] Zhiyuan Wan, David Lo, Xin Xia and Liang Cai. “Bug Characteristics in Blockchain Systems: A Large-Scale Empirical Study,” IEEE/ACM 14th International Conference on Mining Software Repositories, pp.413-424, 2017.

[8] Zhenmin Li, Lin Tan, Xuanhui Wang, Shan Lu, Yuanyuan Zhou and Chengxiang Zhai. “Have Things Changed Now? – An Empirical Study of Bug Characteristics in Modern Open Source Software,” Proc. of the 1st workshop on Architectural and System Support For Improving Software Dependability, pp.25-33, 2006.